Lecture 1

The major components of Central Processing Unit are:

1. Control Unit & Instruction Decoder.
2. Arithmetic / Logic unit
3. Registers

Microprocessor	Microcontroller
Used where the task in intensive	Used for simpler and predefined tasks.
The program can be changed for every application	The program is fixed once it is designed/manufactured.
Only CPU on the chip.	CPU + I/O + Memory on the chip.
Higher clock speed	Lower clock speed
Ram used is higher	Ram used is lower
Power consumption is higher	Power consumption is lower
Cost is higher	Cost is lower
Overall size is greater	Overall size is smaller
Applications include personal PC, laptops, smartphones.	Applications include embedded systems. Such as Keyboard, mouse, washing machine,

For the Following devices, determine whether they use multiprocessor or microcontroller, justify your answer:

i) Keyword - Keyboard is designed for a very small and specific purpose that is to take alphabetical or numeric inputs from the user. It does not require a lot of processing

power or ram to operate as it only has to send signals to the PC regarding the keystrokes. Furthermore keywords are also very low powered and wireless keyboards run on small batteries. All these points to the fact that we use a microcontroller for the keyboard.

ii) Smart Multi-Color LED light - Such lights have a limited functionality of changing color

Lecture 2

System Bus:

- Address Bus
- System Bus
- Control Bus

Address Bus:

- It is UNIDIRECTIONAL
- Always from CPU to I/O or Memory
- Address is generated in the CPU and it carried to the target location which is either in CPU or I/O
- Address bus size is used to determine the maximum memory that can be supported/utilized.

Data Bus:

- It is BIDIRECTIONAL
- From CPU to I/O or Memory & I/O or Memory to CPU.
- Data needs to be transferred from CPU to I/O or Memory (for the writing operation) & and from I/O or Memory to the CPU (for the reading operation)
- Data bus is used to determine the word length

Control Bus:

- Used to send control signals to I/O read and write, memory read and write, Interrupt.
- Signals used to synchronize operations of individual elements.

Q) What is means to be volatile:

- Memories which can store data as long as there is a power supply.
- As soon as the power connection is lost the data is *automatically erased.*
- Volatile memory is temporary
- Eg: RAM, registers

Q) What is ROM:

- Read Only Memory
- Can only be read from, can NOT be written into by the CPU.
- It is *NOT Volatile (Non - Volatile)*
- BIOS (Basic Input/Output System) is loaded onto it along with other preloaded software by the manufacturer
- Can NOT be changed by the user

Q) What is BIOS:

- BIOS (Basic Input/Output System)
- It is the basic level of input / output such as keyboard/ mouse input and monitor output that is accessible right after the startup
- This allows the user to interact with the computer even before Windows is fully booted
- It can not be changed by the user
- It is in ROM

32 - bit CPU:

- Address bus size: 32 bit
- Data bus size: 32 bit

Q) What is the maximum memory(RAM) that can be supported by a 32-bit CPU?

A) To find the maximum memory that can be supported, we need to find the address bus size. For a 32-bit CPU it has an address bus size of 32 bit. So Maximum size supported is 2^{32} = 4.29 x 10^9 Bytes i.e 4 GB (gigaBytes).

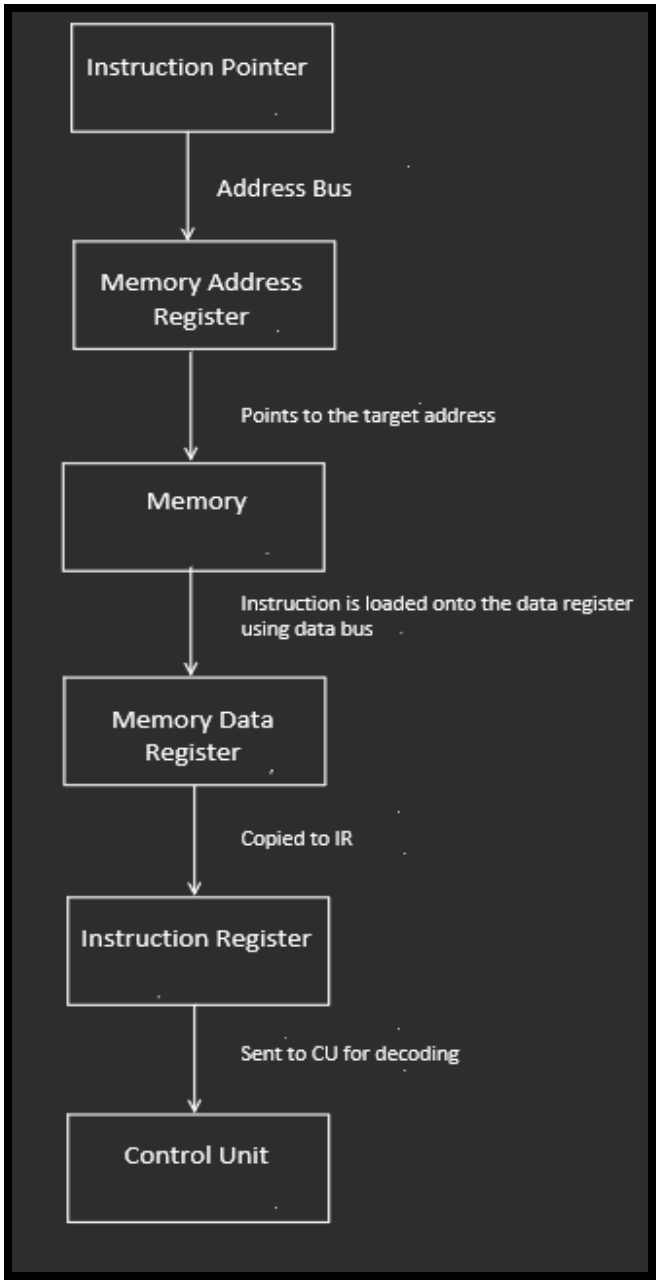
Q) What will happen if we use 16 GB memory with a 32-bit system ?

A) Since the maximum memory that can be supported by a 32 bit system is 4GB. So the rest of the memory after 4GB can NOT be utilized/ addressed/ acknowledged/ supported by the CPU. *For the CPU, the rest of the 12 GB of memory does not exist.*

Q) What will happen if we use 3 GB memory with a 32-bit system ?

A) Since 3GB is below our maximum threshold of 4GB memory. So we can FULLY utilize the 3GB memory.

Q) How does the FETCHING operation works


Instruction Pointer
Address Bus
Memory Address Register
Points to the target address
Memory
Instruction is loaded onto the data register using data bus
Memory Data Register
Copied to IR
Instruction Register
Sent to CU for decoding
Control Unit

Lecture 3

Q) Why do we need segmentation ?

A) The 8086 microprocessor has a 20 bit address bus which allows it to utilize a maximum memory size of 2^{20} = 1 MB (megabytes). Meanwhile it has a data bus of 16 bits which allows it to carry data of size 2^{16} = 64KB (kilobytes). This means to address the entire 1 MB memory there needs to be (1MB/64 kb) = 16 partitions or segments.

```
Logical Address  → Generated by CPU
Physical Address → Actual address in the memory or I/O
```

```
Logical Address = (Segment number : Offset)
```

First / Starting Address of the segment = Segment number x 10h

The last digit(LSB) of the first address of the segment has to be 0.

Physical Address = Starting Address of the segment + Offset

Q) Given the logical address (B214 : 0112)

i) The first address of the segment = B214 x 10h = B2140h

ii) The second address of the segment = B2140h + 1h = B2141h

iii) The offset = 0112

iv) The (target) physical address = B2140h + 0112 = B2262h

v) The last physical address = B2140h + FFFF = C213Fh

vi) The 2nd last physical address = B2140h + FFFF - 1 = C213Eh

Q)Suppose, physical address = 33330h, offset = 0020, calculate the segment no?

A) Physical address = Starting Address of the segment + Offset

Starting Address of the segment = Physical address - Offset

= 33330 - 0020

= 33310h

Starting Address of the segment = Segment number x 10h

Segment number = Starting Address of the segment / 10h

= 33310 / 10 = 3331

Q)Suppose,segment number = CCCC and memory location to be addressed is CCCC2h. What is the offset?

A) Starting Address of the segment = Segment number x 10h

= CCCC x 10h

= CCCC0h

Physical address = Starting Address of the segment + Offset

Offset = Physical address - Starting Address of the segment

= CCCC2h - CCCC0h

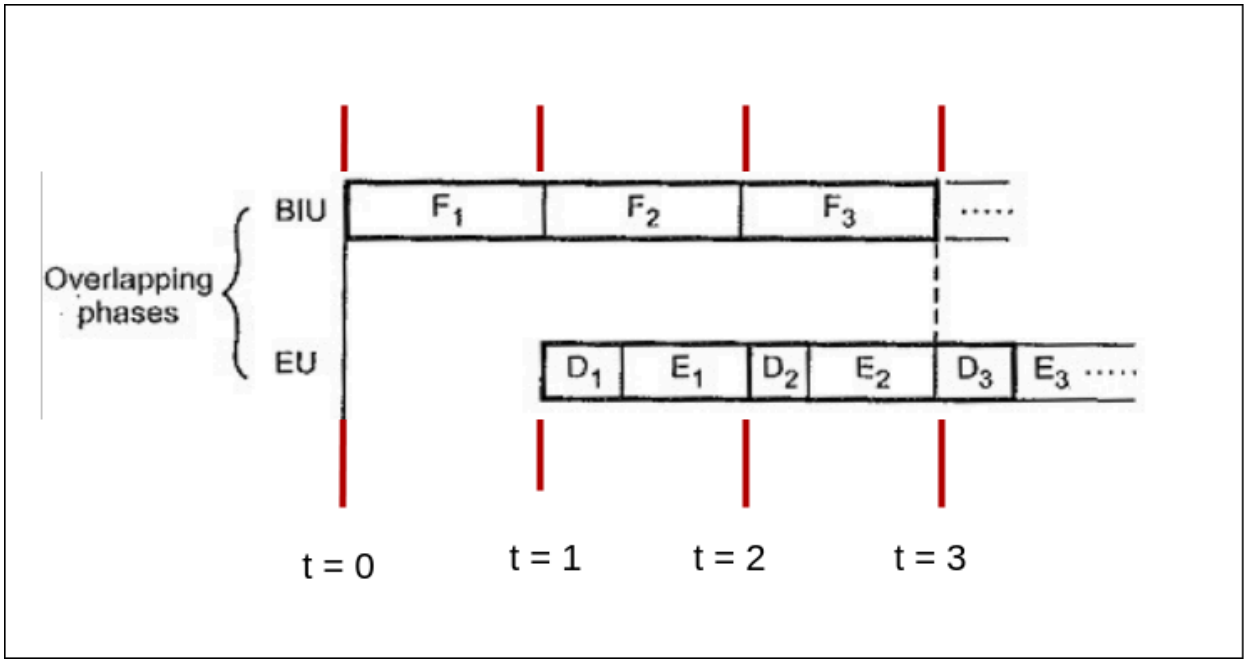
= 0002

Q)Suppose,segment number = CCCC and memory location to be addressed is BCCC2h. What is the offset?

A)

Q) Why does pipelining speed up the execution of the programs in 8086 / How does 8086 take less time to execute the programs compared to its predecessor

A)



Pipeline has two main ideas. One there are two separate units namely BIU (where fetching takes place only) and EU (where decoding + execution takes place). Secondly, there is an instruction queue where instructions are pre-fetched.

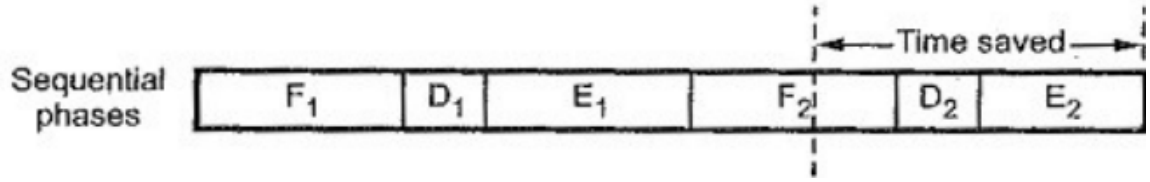
In the time slot t = 0 to t = 1, the INSTRUCTION 1 is being fetched and inserted into the queue.

In the time slot t = 1 to t = 2, the INSTRUCTION 1 is sent from BIU to the EU for decoding and execution. At the same time, the INSTRUCTION 2 is being fetched and inserted in the instruction queue.

In the time slot t = 2 to t = 3, the INSTRUCTION 2 is sent for decoding and execution. At the same time, the INSTRUCTION 3 is being fetched and inserted in the instruction queue.

Q) How would the execution be slower without pipelining?

A)



Without pipelining, there would be no parallelism. Hence first an instruction has to be fetched, then decoded and executed. After the execution of the first instruction is FULLY COMPLETE, only then the second instruction is fetched and then it is decoded and executed.

Q) What does pre-fetching mean?

A) It means the next instruction is being fetched in advance when the current instruction is being executed. It is known as pipelining.

Q) What does the instruction queue is filled up with exactly 2 bytes?

A) 2 bytes = (2 x 8) =16 bits. The size of the data bus is 16 bits. Hence it can transfer exactly 16 bits of data at a time.

Q) Let's say there is 1 byte of free space in the instruction queue. Will a new instruction be fetched?

A) No. There needs to be at least 2 bytes of free space in the instruction queue.

Q) Can we have multiple logical addresses for non overlapping segmentation ?

A) No.